Spreadsheet Does Not = Database

February 3, 2004
By Helen Bradley

Since the days of Lotus 1-2-3, people have used spreadsheet programs for everything from word processing to data management. Doing the former is silly. Doing the latter, however, is viable, especially in the latest version of Microsoft Excel. But though you may be more comfortable with Excel, a real relational database program like Microsoft Access is a better choice for managing data—for a number of reasons.

• Databases are safer. Excel, for example, does everything in memory, so that any unsaved data may be lost if your system crashes. Databases write data to the hard drive immediately.

• Databases can handle more data. Sure, Excel can technically handle more than 65,000 rows of data, but doing so will likely bog down even the fastest PC.

• Databases can easily link tables of related data together, such as customers and orders or musical groups and albums (as well as the songs on each album). This is where the words *relational* and *database* come together. Storing related data together in a single table or spreadsheet can be unwieldy and invite errors.

We'll look at a situation for which Access is a better tool than Excel and show you how an Access solution works. If you've never used Access before, that's okay; we'll walk you through how to create everything from scratch. We used Access 2002 for the instructions, but you'll find the process is similar in all versions of Access. We chose Access because so many users have it already, but you can do the same things in other relational databases such as FileMaker or Microsoft SQL Server. For more on picking the right database, see "Databases for All Reasons" in our issue of January 2003.

More Than a List

Consider a veterinarian's office: To record pet and owner details, you could use a list in an Excel worksheet, but you'd encounter difficulties. If you create one record for each owner, how would you handle an owner with multiple pets? You could add a field for each pet, which would work for most clients. But a client who runs a breeding kennel with 25 cats and innumerable kittens would force your data record to grow to an excessive length.

On the other hand, if you organize your data so you have one record per pet, you would have to enter the owner details for each pet in the household. This is unnecessarily repetitive. And if an owner changes his address, you would have to find and update all his pets' records individually.

The better solution is to have two lists—one with the owner details and one with the pet details—and then link the two by including a field in both lists with a common piece of information. For example, give each owner a unique code number, which you can then use in his pets' records. That way, you can find a pet, check the owner code, and then find the owner's details in the owner file. Likewise, you can look up an owner, find his code, then extract all the pet records with that owner number.

Although you have two lists, each owner and each pet has only one entry in the system. It's neat and efficient, and it solves another problem our veterinarian may encounter: When client breeders sell kittens to new owners, the new owners may become clients, too. To change a pet from one owner to another, simply change the owner code in the pet's record and if necessary add a new owner record.

Create the Database

To create the database requires two tables, one for owners and one for pets, with a field common to both—the owner code. We will set up the relationship between the two tables and add a form to make it easier to enter data.

Each table needs a structure that includes a list of field names and types, as well as the sizes of the fields. Each table must also have a *primary key*—a field that contains a piece of information unique to that record. In the owner's table, the primary key is the owner code; in the pet table, we'll use a similar field called the pet code. We will use an AutoNumber field type for each. Access will then assign a unique sequential value to that field for each record.

To build your database, launch Access, choose *Blank Database* from the task pane, and name your file Vet.mdb. Click on *Create* then double-click on the *Create table in Design view* option. The *Table1:Table* dialog will appear. Type *ClientNo* as the first field name, then tab over to the next column and enter *AutoNumber* as the *Data Type*. (Access automatically completes the entry once you've typed the first letter.) Now enter the rest of the data as shown on the next page. Here are the fields and types:

Field Name	Data Type
ClientNo	AutoNumber
FamilyName	Text
Town	Text
EntryDate	Date/Time

If you want, you can add a description for each field to explain its contents as well as a caption. The caption is a name that is used in place of the field name in reports and forms. If you use shortened or cryptic field names, captions are a good idea.

To set a primary key, right-click on the area to the left of the *ClientNo* field and choose *Primary Key*. A key icon will appear, indicating that the field is the primary key. Save the file with the name *Clients,* and click on the table's *Close* button.

Repeat this process to create a second table for pets with these fields:

Field Name	Data Type
PetID	AutoNumber
ClientNo	Number
Type	Text
Name	Text

Set *PetID* as the primary key, name the table *Pets,* and close it.

Once you have created the tables, you can define the relationship between them. When you do this, Access helps you maintain your data integrity. For example, you can set up the relationship so that removing an owner automatically removes any of his pets from the *Pets* table.

Choose *Tools | Relationships.* When the *Show Table* dialog appears, click on the *Clients* table and then select *Add.* Do the same with the *Pets* table and then click on *Close.* Small dialogs will appear, showing the structure of the two tables. Drag the *ClientNo* field from the *Client* table and drop it on the *ClientNo* field in the *Pets* table. When you let go of the mouse button, the *Edit Relationships* dialog appears with these two fields listed. Select the *Enforce Referential Integrity* check box and the *Cascade Delete Related Records* check box. This ensures that if an owner is removed, all his pets are removed, too. Click on *Create* to set up the relationship, which is one-to-many—one owner can have many pets (Figure 1). Click on the window's *Close* button and

answer *Yes* when prompted to save the changes.

Now you can enter data into the tables. Click on *Tables* in the *Objects* bar and double-click on *Clients* to open it in datasheet view. Type the following data into the table (the number in the *ClientNo* field will be entered automatically):

CLIENTNO	NAME	TOWN	ENTRY DATE
1	Brown	Athens	12/30/1998
2	Smith	Athens	2/2/2000
3	Green	Atlanta	5/5/2000

Close the table and then repeat the process to add the following data to the *Pet* table (the *PetID* will be added automatically):

PetID	ClientNo	Type	Name
1	2	Cat	Peaches
2	1	Dog	Sam
3	3	Horse	Dobbin
4	3	Cat	Ginger

Create a Data Entry Form

Although you could continue to add data using the two tables separately, it's easier to use a form that displays all the related data. Access can do this for you. Close both tables and click on the *Forms* icon in the *Objects* bar and double-click on *Create form by using wizard.*

From the *Tables / Queries* drop-down list choose *Table:Clients* and click on the double angle brackets (>>) to move all the *Available Fields* to the *Selected Fields* pane. Then choose *Table:Pets* and move only the *Type* and *Name* fields from the *Available Fields* to the *Selected Fields* pane. Then click on *Next.*

Access will ask you, *How do you want to view your data?* Choose *by Clients* and click on the *Form with subform(s)* option and then choose *Next.* When prompted, select *Datasheet* as the layout type for the subform and choose *Next.* Pick a style for your form (any will do) and click on *Next.* Type a form name, such as *Client and Pet details,* click on *Open* to view or enter information in the form and click on *Finish* to end.

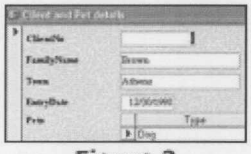

Figure 2

ENLARGE

A form appears on the screen with the client data on top and the details of the pets belonging to the client in a table below (Figure 2). You'll see two sets of record navigation tools. The one at the bottom of the table is for the *Pets* subform and the other is for the client records. Click on the *Next Record* button for the *Client* data and you will see that two pets are displayed for Record 3.

Now you can add a new client and his pet, as well as add a new pet to one of the existing clients. To see what is happening behind the scenes, close the form and open the *Pets* table. You'll see that the data has been entered into the fields *PetID* and *ClientNo,* even though neither field was included on the form. The *PetID* number is automatically entered, because the field type is *AutoNumber* and the *ClientNo* field is added automatically, since the records are related through the form's design.

Remove a client from the *Clients* table by opening the table, selecting the client, and clicking on *Delete.* You'll be warned that a record in another data file will be affected (the client's pets will be removed when the client is). This is the result of selecting the *Cascade Delete Related Records* check box when setting up the

relationship. The same does not work in reverse and it is possible to have a client with no pets in the *Clients* table.

Our scenario is a simplified one to help you see how you can set up a relational database. To learn more, check out the database section of your local library or bookstore. A good place to start is with a book such as *Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design* by Michael J. Hernandez.